Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 95 (2016) 87 – 94

Procedia Computer Science

Complex Adaptive Systems, Publication 6
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2016-Los Angeles, CA

SoS Meta-Architecture Assessment by Dual Application of Rule Based Fuzzy Inference Systems

Andrew Renault*

Missouri University of Science and Technology, Rolla, MO-65409, USA

Abstract

In the early stages of system of systems (SoS) meta-architecture concept development, it is greatly beneficial to have a formal method of architecture assessment to ensure compliance to stakeholder needs and requirements. Key performance attributes (KPAs) and key functional attributes (KFAs) are linguistic terms that represent the non-functional and functional requirements of the architecture. KPAs and KFAs can be assessed and converted into numeric values that can be used as fuzzy inputs to Mamdani-type rule based fuzzy inference systems (MRBFIS) to make fuzzy assessments of SoS meta-architecture concepts. This paper details an application of an assessment model featuring the integration of two separate MRBFISs; each with five inputs, five membership functions (MFs), twenty five If-Then rules, and a single output that is determined by the fuzzy rule base. Each unique MRBFIS used in the assessment was developed using the MATLAB Fuzzy Logic Designer application. The numeric outputs of each MRBFIS were then combined to provide an integrated KPA and KFA total architecture assessment score. The assessment model detailed in this paper provides a formal method of architecture assessment to be used in conjunction with trade space exploration, systems scoping, and the concept generation phase. This assessment model may help prevent costly project failures, system ambiguity late in development, unacceptable risks, cost overrun, and the inability to develop critical systems.

Peer-review under responsibility of scientific committee of Missouri University of Science and Technology

Keywords: rule based fuzzy inference systems; formal methods for systems architecting; architecture assessment

1. Introduction

The systems architecting generation process involves several stages that lead to the development of a conceptual systems architecture. The development of a systems of systems (SoS) meta-architecture requires a preliminary method for exploring conceptual architecture trade space [1] to find concepts more in compliance to customer

* Corresponding author. Tel.: 1-425-328-1013
E-mail address: andrew11@engineer.com

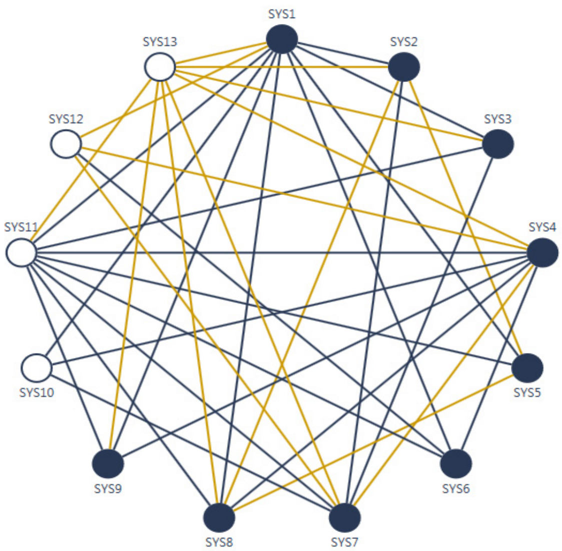
Peer-review under responsibility of scientific committee of Missouri University of Science and Technology
doi:10.1016/j.procs.2016.09.297

requirements as they are understood at this stage. The systems architecting process includes systems scoping, aggregating, partitioning, integrating and validating the systems architecture prototype. The architecting process is the process by which standards, protocols, rules, system structures, and interfaces are created in order to achieve the functional and non-functional requirements of a system [2]. It is essential that the SoS meta-architecture be assessed by the systems architect in the very early stages of development and it is beneficial to have a formal method of assessing the concept. A formal assessment model should include both functional and non-functional requirements of the concept to adequately describe stakeholder needs and expectations and assess concepts against them.

There are a wide range of possibilities for assessing conceptual architectures in the early developmental stages. Key performance attributes (KPAs) and non-functional requirements can effectively represent the customer needs and capabilities of SoS meta-architectures. KPAs are also represented by linguistic terms called *Ilities* [3] or they may also be called quality attributes. Key functional attributes (KFAs), as introduced in this paper, are linguistic terms for the major functional requirements that enable the overall SoS Meta-architecture capabilities. There is a current lack of formal methods of architecture assessment [4, 5]. A preliminary overall assessment by suitable KPAs and KFAs is beneficial in the early stages of architecture development to identify system ambiguity and to determine if the concept is feasible. Furthermore, KPAs and KFAs that represent the architecture concept can be used as inputs to dual Mamdani-type rule based fuzzy inference systems (MRBFISs) to provide an additional assessment.

1.1. Systems of Systems (SoS) Meta-Architecture Concept Generation

The systems architecting generation process involves several stages that lead to the reduction of system ambiguity and the effective use of the trade space assignments. The SoS meta-architecture is the conceptual model that defines the structure, behaviour, and more views of the SoS. The architecture concept generation phase is initiated after the customer needs have been assessed and validated. The architecture concept may evolve, adapt, and improve throughout the generation stages until an acceptable architecture is realized. The selected KPAs and KFAs must accurately represent the architecture concept to provide an indication of whether to proceed further in development. Figure 1a represents the network diagram of the SoS meta-architecture concept (components and interfaces) with the upper triangular matrix representation of the system interfaces shown on Figure 1b. This paper will make an assessment on this conceptual SoS meta-architecture using five KPAs and five KFAs that represent the overall system requirements and capabilities.



	SYS1	SYS2	SYS3	SYS4	SYS5	SYS6	SYS7	SYS8	SYS9	SYS10	SYS11	SYS12	SYS13
SYS1	1	1	1	0	1	1	0	1	1	1	1	1	1
SYS2		1	0	0	1	0	1	1	0	0	0	0	1
SYS3			1	0	0	0	1	0	0	0	1	0	1
SYS4				1	0	1	1	1	1	1	1	1	1
SYS5					1	0	0	1	0	0	1	0	0
SYS6						1	0	0	0	0	1	1	0
SYS7							1	0	0	1	1	1	1
SYS8								1	0	0	1	0	1
SYS9									1	0	1	0	1
SYS10										0	0	0	0
SYS11											0	0	1
SYS12												0	0
SYS13													0

Figure 1. (a) SoS meta-architecture network diagram (b) SoS meta-architecture upper triangular matrix

1.2. Assessment Criteria using both Non-Functional and Functional Requirements

Key performance attributes (KPAs) may include non-functional requirements that specify criteria used to evaluate the operation of the SoS meta architecture rather than specific behaviors or functions. Non-functional KPAs are also called *"Ilities"* [3, 6] or quality attributes. KPAs are also non-functional requirements that cover all the remaining requirements which are not covered by the functional requirements. There are a numerous KPAs that could be selected and the selected candidates will represent the architectures primary capabilities. For system to effectively document the requirements for KPAs, and to implement and verify such properties, the language and translated assessment must be precise enough to support future engineering activities. KPAs are considered in the design of the system architecture because they represent significant customer needs.

Key functional attributes (KFAs) are functional requirements that define specific behaviours or functions. A KFA is a functional requirement that essentially specifies something that the system should do. KFAs are architecturally significant functional requirements that are detailed in the functional and systems architecture. The usage of KFAs as architecture assessment criteria introduces a new assessment approach. When both KPAs and KFAs are used to evaluate an architecture concept, a more thorough overall assessment is possible. The systems architect can make an individual assessment score for each KPA and KFA to be used as fuzzy inputs to dual MRBFISs. A total of five KPAs and five KFAs, as shown on Figure 2a and 2b, will be chosen to represent the architecture concept as these quantities are ideal inputs to the MRBFIS, as too many attributes may complicate the assessment.

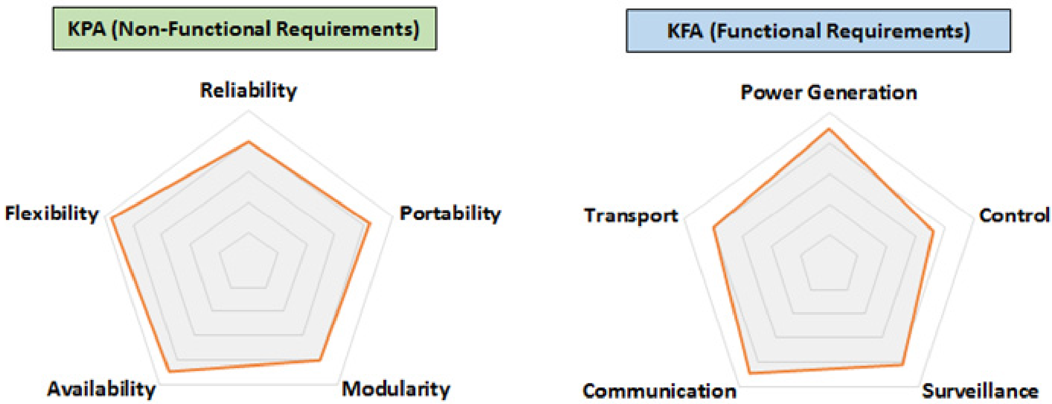


Figure 2. (a) Kiviat chart of the KPAs (b) Kiviat chart of the KFAs

1.3. Determination of KPA and KFA Individual Scores.

The section provides a brief description on how individual KPA and KFA scores are generated once they are selected. The systems architect will decide how to assess each individual attribute to generate individual scores that are to be used as dual MRBFIS inputs. A total of five KPA scores and five KFA scores (number between 0 & 1) will be determined. This process will be explained in more detail in Section 2 and is illustrated in Figure 7.

Trade studies, matrix diagrams, and risk assessment matrices are effective tools [7] in evaluating the non-functional requirements of the conceptual SoS-meta architecture. System architects often rely on past experience using a heuristic assessment approach [8]. Functional modeling, design structure matrix (DSM) [9, 10], and quality function deployment (QFD) [11, 12] are effective tools in evaluating the functional requirements of the architecture. A thorough assessment of the functional requirements using checklists is also beneficial.

A systems architect can utilize mission scenarios [13, 14] as a simulated assessment tool to evaluate the architecture concepts perceived functionality. The KPAs and KFAs used in the examples shown in Section 3 are arbitrary and calculation methods are not provided for individual scores. This paper introduces a new architecture assessment approach that utilizes the KPA and KFA assessment scores and dual MRBFISs to add adjustable nonlinear assessment capabilities to the final assessment output. Figure 3a and 3b are examples of assessment criteria that may be used collectively to evaluate individual KPAs and KFAs to generate numeric scores.

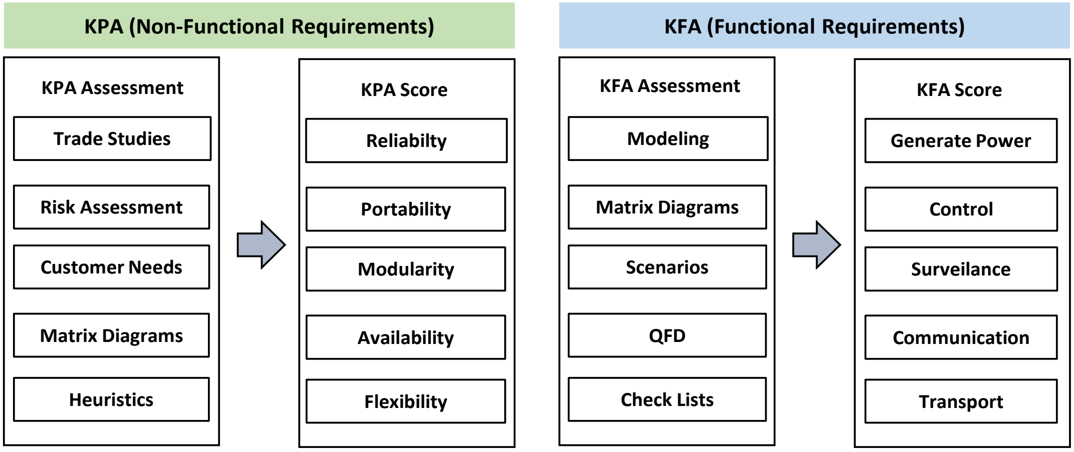


Figure 3. (a) Assessment of the KPAs (b) Assessment of the KFAs

1.4. Fuzzy Evaluations of the KPAs and KFAs

Each of the five KPA and KFA input variables assigned to the dual MRBFISs have five membership functions (MFs) scaled from “lowest” to “highest” as shown on Figure 4a. The output variables assigned to both of the MRBFISs have five MFs scaled from “lowest” to “highest” as illustrated on Figure 4b. Let *U* be a *universe of discourse* for all fuzzy sets. A fuzzy value on *U* is characterized by a fuzzy set *F* in *U*. A membership function that is expressed as μ_F: $U \rightarrow [0, 1]$ is defined for all of the fuzzy ranges. Increasing the number of MFs within each input and output variable is beneficial to provide a crisp output as compared to having fewer MFs. Decreasing the number of MFs increases the fuzziness of the FIS output while increasing the number of MFs increases the output crispness.

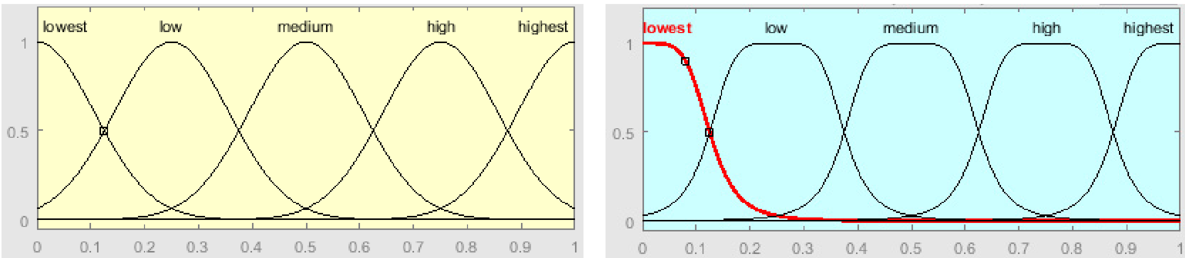


Figure 4. (a) Generalized bell curve MF (b) Gaussian rounded trapezoidal curve MF

The KPA MRBFIS (Figure 5a) and KFA MRBFIS (Figure 5b) each contained twenty five individual If-Then rules. The rules selected logic AND as the connection between all input KPA and KFA variables. The output of the MRBFIS is dependent on the MFs settings and the fuzzy If-Then rules. It is possible to have the “medium” MRBFIS output even though four KPA input variables scored in the “high” MF input range as shown in Rule 13 below.

Rule 13: If (Reliability is medium) and (Portability is high) and (Modularity is high) and (Availability is high) and (Flexibility is high) then (output1 is medium)

The rules when combined with the MFs can assist in obtaining a crisp MRBFIS output and allow a nonlinear assessment that is not possible with a sum or average of individual scores. The MRBFIS generated output surface examples as shown in Figures 6a and 6b illustrate the ability of the MRBFIS to produce adjustable nonlinear relationships between KPAs and KFAs. The MRBFIS is very effective when used as part of an architecture assessment tool because the outputs are based on the MFs and the rule based relationships between attributes.

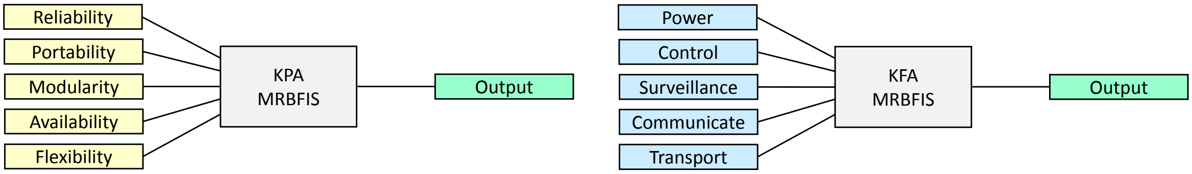


Figure 5. (a) KPA MRBFIS inputs and output (b) KFA MRBFIS inputs and output

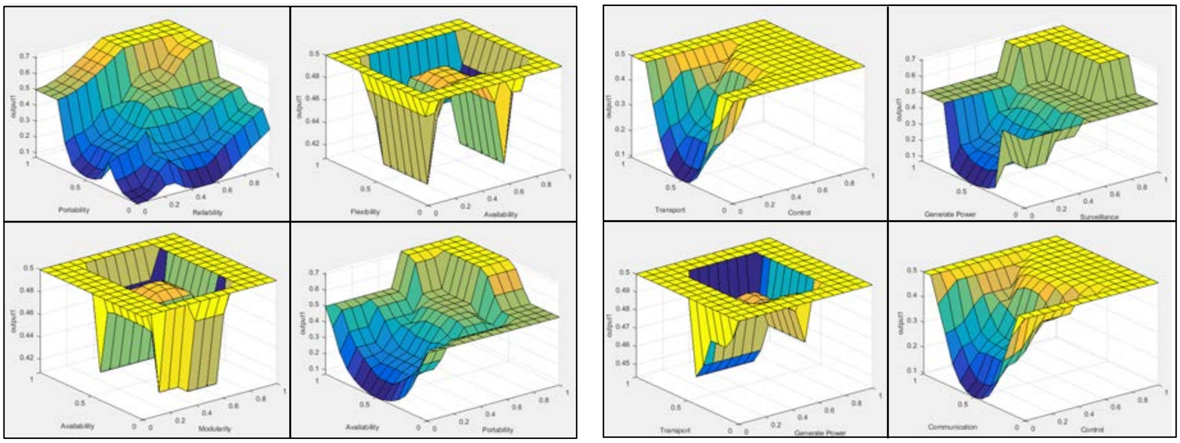


Figure 6. (a) KPA MRBFIS output surfaces (b) KFA MRBFIS output surfaces

2. Proposed Model for the SoS Meta-Architecture KPA and KFA Integrated Assessment

The SoS meta-architecture KPA and KFA assessment process and model as illustrated on Figure 7 is used to assess the architecture (see Figure 1a) at the early stages of conceptual development. The process begins with the assessment of individual KPAs and KFAs. The KPA (non-functional requirements) assessment includes five system engineering tools [7] that include; trades studies, risk assessments, customer needs assessments, matrix diagrams, and design heuristics. Each individual KPA is evaluated using the five assessment categories and a score is determined (between 0 and 1) and recorded. The KFA (functional requirements) assessment includes five system engineering tools [7] that include; modeling, matrix diagrams, scenarios, quality function deployment (QFD), and check lists. Each individual KPA is evaluated using the five assessment categories and a score is determined (between 0 and 1) and recorded. The KPA scores are entered as inputs into the KPA MRBFIS and the KFA are entered as inputs to the KFA MRBFIS as shown on Figure 7.

The fuzzy inference section of the assessment model is the process that formulates the mapping from five given inputs to an output using fuzzy logic and If-Then rules. The mapping of the inference section then provides a basis from which decisions can be made, or patterns discerned. The process of fuzzy inference involves all of the formulated integrated controls contained in the MFs (see Figure 4), logical operations, and the twenty five If-Then rules. The fuzzy If-Then rules are expressions of the form “If X Then Y”, where X and Y are labels of fuzzy sets characterized by appropriate MFs. The set of If-Then rules relate to a fuzzy logic system that are stored together and called a fuzzy rule base. The KPA MRBFIS and the KFA MRBFIS each have their own unique fuzzy rule base.

The final stage of the assessment model is called the defuzzifier and this stage converts the fuzzy sets inputted into each MRBFIS into two crisp outputs that are either a linguistic or numeric value. The defuzzifier stage then combines the dual MRBFIS crisp outputs into a single numeric value that represents the total assessment score to determine if the architecture concept shown in Figure 1a is acceptable or unacceptable.

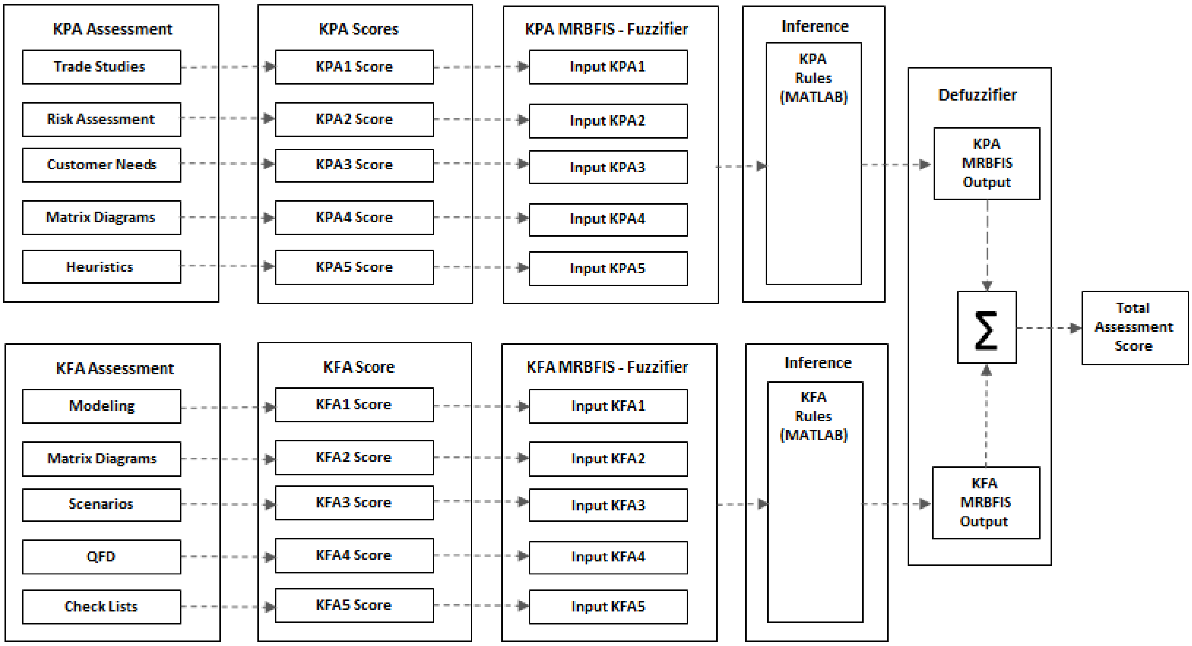


Figure 7. SoS Meta-Architecture KPA and KFA Assessment Process and Model.

3. Application of the Assessment Proposed Model

This section demonstrates an application of the assessment model using assumed values for the KPA and KFA MRBFIS inputs to serve as an example of its effectiveness. The determination of KPA and KFA individual scores is briefly described in Section 1.3 without going into detailed calculation methods as this is up to the discretion of the systems architect. The objective of this assessment model is to add the element of fuzzy If-Then rules to both KPA and KFA assessment score inputs to provide a comprehensive overall formal architecture assessment model (see Figure 7). The fuzzy If-Then rules of each MRBFIS allows inputs to be prioritized and combined to enable a customized output that is more adjustable than other assessment methods.

It is predicted that this assessment model will provide a unique multi-dimensional approach to formal architecture assessment methods. The model is able to assess the SoS meta-architecture once MRBFIS five inputs are supplied and MFs and If-Then rules are configured. Total architecture assessment scores of 0.80 (80%) and above will be considered as acceptable. The required 80% total assessment score is an arbitrary value that used for the example in this paper. It should be noted that the assessment model could also be used with lower or higher total assessment scores as decided by the systems architect.

The following is an example of the KPA assessment using MATLAB® code to run the MRBFIS fuzzy assessor:

```
fismat = readfis('KPA MRBFIS');
out1 = evalfis([.90 .87 .62 .67 .63],fismat)
Select RUN from the MATLAB® menu to obtain assessment:
out1 = 0.804
```

The KPA individual input assessments and the results of the KPA MRBFIS output assessment are detailed in Figure 8 below. The output of the KPA MRBFIS is 0.804 (80.4%) while the average of the KPA inputs is 0.74 (74%). This is a example of how the rule base can allow a higher MRBFIS output than the average of the inputs.

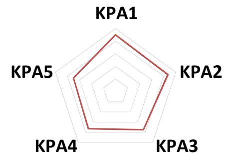
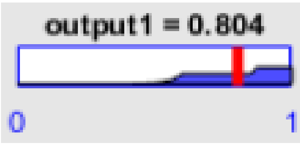
<table>
<tr><th colspan="5">INPUT</th><th>OUPUT</th></tr>
<tr><th>KPA Kiviat Chart</th><th>KPA</th><th>KPA Name</th><th>FIS Input</th><th>Avg</th><th>MATLAB MRBFIS</th></tr>
<tr><td rowspan="5">
</td><td>1</td><td>Reliability</td><td>0.90</td><td rowspan="5">0.74</td><td rowspan="5">
</td></tr>
<tr><td>2</td><td>Portability</td><td>0.87</td></tr>
<tr><td>3</td><td>Modularity</td><td>0.62</td></tr>
<tr><td>4</td><td>Availability</td><td>0.67</td></tr>
<tr><td>5</td><td>Flexibility</td><td>0.63</td></tr>
</table>

Figure 8. SoS Meta-Architecture KPA Assessment using MATLAB® code to run the MRBFIS.

The following is an example of the KFA assessment using MATLAB® code to run the MRBFIS fuzzy assessor:

fismat = readfis('KFA MRBFIS');
out1 = evalfis([.90 .88 .84 .87 .80],fismat)
Select RUN from the MATLAB® menu to obtain assessment:
out1 = 0.809

The KFA individual input assessments and the results of the KFA MRBFIS assessment are detailed in Figure 9 below. The output of the KFA MRBFIS is 0.809 (80.9%) while the average of the KFA inputs is 0.86 (86%). This is a good example of how the rule base can allow MRBFIS output lower than the average of the inputs.

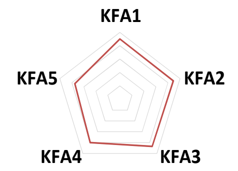
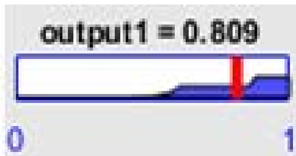
<table>
<tr><th colspan="5">INPUT</th><th>OUPUT</th></tr>
<tr><th>KFA Kiviat Chart</th><th>KFA</th><th>KFA Name</th><th>FIS Input</th><th>Avg</th><th>MATLAB MRBFIS</th></tr>
<tr><td rowspan="5">
</td><td>1</td><td>Control</td><td>0.90</td><td rowspan="5">0.86</td><td rowspan="5">
</td></tr>
<tr><td>2</td><td>Surveillance</td><td>0.88</td></tr>
<tr><td>3</td><td>Generate Power</td><td>0.84</td></tr>
<tr><td>4</td><td>Communication</td><td>0.87</td></tr>
<tr><td>5</td><td>Transport</td><td>0.80</td></tr>
</table>

Figure 9. SoS Meta-Architecture KFA Assessment using MATLAB® code to run the MRBFIS.

The total architecture score is an average of the KPA and KFA MRBFIS numeric outputs (80.4% and 80.9%) and in the case of this example is 80.65%, which is an acceptable total architecture score exceeding 80%.

4. Conclusions and Future Work

The assessment model detailed in this paper provides a formal method for assessing SoS meta-architectures during the trade space exploration and concept generation phases. The first stage of the assessment model effectively utilizes KPA and KFA linguistic classifiers to represent the non-functional and functional requirements and then translates them into a numeric values in relation to how well they meet customer requirements and expectations. The second stage of the assessment model inputs the fuzzy KPA and KFA scores into the dual MRBFISs to provide inference with five MFs, a fuzzy rule base consisting of twenty five If-Then rules, and fuzzy logic. The final stage of the assessment model provides defuzzification by combining the dual MRBFISs numeric outputs into a total architecture score that represents the SoS meta-architectures non-functional and functional characteristics. The fuzzy IF-Then rules of the dual MRBFISs enables a nonlinear assessment that is not possible with the sum or average of the individual attributes. This assessment model may prove to be an effective assessment tool and may help prevent potential project failures, system ambiguity late in development, unacceptable risks, and the inability to develop critical systems. Future work may explore more detailed methods of calculating the numeric values for the individual KPA and KFA scores to be used as inputs to the MRBFISs.

References

1. Pape L, Agarwal S, Dagli C. *Selecting Attributes, Rules, and Membership Functions for Fuzzy SoS Architecture Evaluation*. Procedia Computer Science, Volume 61, 2015, pp. 176-182.
2. De Weck OL, Roos D, Magee CL. *Engineering Systems: Meeting Human Needs in a Complex Technological World.* Massachusetts: MIT Press, 2013.
3. Dou K, Wang X, Tang C, Ross A. *An Evolutionary Theory-Systems Approach to a Science of the Ilities.* Procedia Computer Science, Volume 44, 2015, pp. 433-442.
4. Renault A. *A Model for Assessing UAV System Architectures*. Procedia Computer Science, Volume 61, 2015, pp. 160-167.
5. Rodano M, Giammarco K. *A Formal Method for Evaluation of a Modeled System Architecture*. Procedia Computer Science, Volume 20, 2013, pp. 210–215.
6. Nguyen CH, Pedrycz W, Duong TL, Tran TS. *A genetic design of linguistic terms for fuzzy rule based classifiers*. International Journal of Approximate Reasoning 54, 2013, pp. 1–21.
7. Goldberg BE, Everhart K, Stevens R, Babbitt N, Clemens P, Stout L. *System Engineering "Toolbox" for Design-Oriented Engineers*. NASA Publication Report 1358, MSFC, 1994.
8. Daly SR, Yilmaz S, Christian AL, Seifert CM, Gonzalez R. *Design Heuristics in Engineering Concept Generation*. Journal of Engineering Education, Vol. 101, 2012, pp. 601–629.
9. Sharon A, De Weck OL, Dori1 D. *Model-Based Design Structure Matrix: deriving a DSM from an Object-Process Model.* In Second International Symposium on Engineering Systems, 2009, pp. 1-12
10. Bartolomei JE, Hastings DE, De Neufville R, Rhodes DH. *Engineering Systems Multiple-Domain Matrix: An Organizing Framework for Modeling Largescale Complex Systems.* Systems Engineering 15.1, 2012, pp. 41–61.
11. Verma R, Maher T, Pullman M. *Effective product and process development using quality function deployment*. Articles 561, 1998.
12. Burge S. *A Functional Approach to Quality Function Deployment.* Technical Paper 0001/sb, 2007.
13. Sutcliffe AG, Maiden NAM, Minocha S, Manuel D. *Supporting Scenario-Based Requirements Engineering.* IEEE Transactions on Software Engineering, Volume. 24, No. 12, 1998.
14. Hallikas J, Pynnönen M, Savolainen P. *Scenario-based approach for evaluating service concepts in the ICT sector*. Proceedings of the 42nd Hawaii International Conference on System Sciences, IEEE, 2009.
15. Ross AM, Rhodes DH. *Towards a Prescriptive Semantic Basis for Change-type Ilities*. Procedia Computer Science, Volume 44, 2015, pp. 443-453.